Normalization without reducibility

René David

Normalization without reducibility

René DAVID*

Abstract

In [8], general results (due to Coppo, Dezani and Veneri [5], [6]) relating properties of pure λ terms and their typability in some systems with conjunctive types $D\Omega$ and D are proved in a uniform way by using the reducibility method. This paper gives a very short proof of the same results (actually, one of them is a bit stronger) using purely arithmetical methods.

MSC : 03B40, 03F05

Keywords : λ-calculus, normalization

1 Introduction

In [8], Gallier presents a uniform approach for proving general results relating properties of pure λ terms and their typability in some systems with conjunctive types $D\Omega$ and D, due to Coppo, Dezani and Venneri ([5],[6]). Gallier's approach uses the reducibility method. The results are not new but the accent is put on the uniformity of the various proofs. Other proofs of similar results can also be found in [1], [15] or [11]. Bucciarelli & al show in [4] that the strong normalization of system D can be easily derived from the one of the simply typed λ-calculus.

I give here another proof of the same results as in [8] (cf theorem 6). Actually the point 4 of theorem 6 is stronger (and this result is new) than the corresponding one in [8] : For an unsolvable term, I give a precise relation between the arity of its type and the number (up to reduction) of its leading λ abstractions.

This proof does not use reducibility and is purely syntactic. The main idea is given at the beginning of section 3.2. It is also completely uniform, very short and (at least in my mind) ... elegant. I also believe that this proof should help to better understand the relations between pure λ-terms and the systems D and $D\Omega$. Note that a very elementary and short (i.e. a half page) proof of the strong normalization of the system D can be "extracted " from this paper. It uses only the part of lemma 18 concerned with SN and the (trivial) lemma 12.

*René David. Laboratoire de Maths. Campus Scientifique. F-73376 Le Bourget du Lac. email david@univ-savoie.fr

It is known that, for fundamental reasons, the technique used below cannot be extended to, for example, the system F. I do not know neither how to extend it to, for example, Gödel's system T nor if, for some reason, no extension should exist.

Acknowledgments The proof of the part (1) of theorem 6 is a very simplified version of the proof given by R Matthes in an informal talk in the Logic Meeting in Oberwolfach (January 1998) after which I had helpful discussions with him. Thanks to K Nour for helpful comments and to the anonymous referee who did a very careful reading of the paper and pointed out many imprecisions.

2 The theorem

I assume the basic notions on pure and typed λ calculus are known. They can be found in any text book on the subject (for example : [10], [2], [9]). For the sake of completeness I recall some notations and the definitions concerning the systems D and $D\Omega$.

$t \to t'$ (resp $t \twoheadrightarrow t'$) means that t reduces to t' by one step (resp some steps, possibly 0) of β reductions.

Every λ term can be uniquely written as $\vec{\lambda}\ (R\ \vec{u})$ where $\vec{\lambda}$ is a (possibly empty) sequence of abstractions, R is either a redex (called the *head redex*) or a variable (in this case the term is said to be in *head normal form* and the variable is the *head variable*) and $\vec{u}$ is a (possibly empty) sequence of arguments.

The *head reduction* consists in reducing the head redex. The *left reduction* consists in reducing the head redex (if there is one) or (inductively) in doing the left reduction of the arguments of the head variable. A (finite or infinite) reduction $t_0 \to t_1 \to ...$ is a *quasi head* (resp *quasi left*) reduction if for every j the reduction $t_i \to t_{i+1}$ is a head reduction (resp a left reduction) for some $i \geq j$.

$cxty(t)$ represents the complexity of t, i.e. the number of symbols occurring in t. If $\vec{u}$ is a sequence of terms and N is a set, $\vec{u} \in N$ means that every element of the sequence $\vec{u}$ is in N.

There are two ways of presenting the *conjunctive* types. (For a history of the different formulations, see [14])

- The first one (see [3]) is the following : The types are constructed from a set of base types and the undefined type ω, using the type constructors $\to$ and $\cap$. The typing rules of the system $D\Omega$ are the following :

(ax)	$\Gamma, x : A \vdash x : A$ and $\Gamma \vdash t : \omega$
$(\to_i)$	If $\Gamma, x : A \vdash t : B$ then $\Gamma \vdash \lambda x\ t : A \to B$
$(\to_e)$	If $\Gamma \vdash u : A \to B$ and $\Gamma \vdash v : A$ then $\Gamma \vdash (u\ v) : B$
$(\cap_i)$	If $\Gamma \vdash t : A$ and $\Gamma \vdash t : B$ then $\Gamma \vdash t : A \cap B$
$(\cap_e)$	If $\Gamma \vdash t : A \cap B$ then $\Gamma \vdash t : A$ and $\Gamma \vdash t : B$

The system D is obtained by restricting the system $D\Omega$ to ω-free types (i.e. types where ω does not occur) and by deleting the axiom $\Gamma \vdash t : \omega$.

- The previous way of presenting $D\Omega$ causes some problem because $(A \to B \cap C)$ is less convenient than $(A \to B) \cap (\ A \to C)$.

  The second way (see [5], [6]) is thus the following : We restrict the set of types by forbidding $\cap$ after $\to$. More precisely, the set T of types and the set S (of *regular* types) are defined by the following grammars (where V is the set of base types) :

$$S = V \mid \omega \mid T \to S \text{ and } T = S \mid S \cap T$$

The typing rules are the same as in the first presentation but the types occurring in the rules must be in T, i.e. in the rules $A \in T$ and $B \in S$.

The following result shows that the two presentations are essentially the same.

Definition 1 *The translation* * *from* $D\Omega$ *into* T *is defined by :*

- *For* $a \in V \cup \{\omega\}$, $a^* = a$.
- $(A \cap B)^* = A^* \cap B^*$.
- $(A \to B)^* = \bigcap (A^* \to B_i)$ *where* $B^* = \bigcap B_i$ *and, for every* i, $B_i \in S$.

Proposition 2 *1. If* $\Gamma \vdash_T t : A$, *then* $\Gamma \vdash_{D\Omega} t : A$.

2. If $\Gamma \vdash_{D\Omega} t : A$ *then* $\Gamma^* \vdash_T t : A^*$.

Proof. Immediate, by induction on the length of the typing derivation. ■

Notations

- I will use the second presentation which is, for my purpose, more convenient. Every type mentionned in the rest of this paper is thus assumed to be in T. In particular, $\Gamma \vdash t : A$ means that A and the types in Γ are in T and that t has type A in the context Γ.
- If $\Gamma = \{x_1 : A_1,\ ...,\ x_n : A_n\}$ and $\Delta = \{x_1 : B_1,\ ...,\ x_n : B_n\}$ the context $\{x_1 : A_1 \cap B_1,\ ...,\ x_n : A_n \cap B_n\}$ will be denoted by $\Gamma \cap \Delta$.

Definition 3 *1.* $t \in SN$ *iff* t *is strongly normalizing.*

2. $t \in WN$ *iff* t *is weakly normalizing.*

3. $t \in HN$ *iff* t *is solvable (i.e.* t *reduces to a head normal form).*

4. For $k \geq 1$,

   (a) $t \in H_k$ *if* t *begins with* k *many* λ.

   (b) $t \in WHN_k$ *iff either* $t \in HN$ *or* t *reduces to a term in* H_k.

Definition 4 *Let* A *be a regular type.*

1. *$A \in WD$ if it is ω-free.*

2. *$A \in HD$ if it is non trivial i.e. $A = A_1 \to ... \to A_n \to a$ for some $a \in V$ and $n \geq 0$.*

3. *For $k \geq 1$, $A \in WHD_k$ iff $A \in HD$ or A has the form $A_1 \to ... \to A_k \to \omega$.*

Examples

Let o be a base type. Then : $(o \to o) \in WD. (\omega \to o) \in HD - WD$. $(o \to o \to \omega) \in WHD_2 - HD$.

Definition 5 *1. $t \in ST$ iff t is typable in D.*

2. *$t \in WT$ iff $\Gamma \vdash t : A$ for some $A \in WD$ and some Γ such that the types in Γ are conjunctions of types in WD.*

3. *$t \in HT$ iff $\Gamma \vdash t : A$ for some $\Gamma \in T$ and $A \in HD$.*

4. *For $k \geq 1$, $t \in WHT_k$ iff $\Gamma \vdash t : A$ for some $\Gamma \in T$ and $A \in WHD_k$.*

Examples and comments

1. It is easy to check that the previous definitions (ST, WT and HT) correspond to the ones in [8]. For example, $t \in HT$ iff t is typable in (the original) $D\Omega$ with a non trivial type (in the sense of [8]).

2. Note that, in 3 and 4, there is no condition on Γ. It is easy to check that 2 (resp. 3) means that the closure of t is typable (in the empty context) of a type in WD (resp. HD).

3. Let $t = ((\lambda x.y)\ (\delta\ \delta))$ where $\delta = \lambda x.\ (x\ x)$. Since $y : o \vdash t : o$ (where x is given the type ω), $t \in WT$.

4. Let $t = \lambda x.\ (x\ (\delta\ \delta))$. Since $\vdash t : (\omega \to o) \to o$, $t \in HT$.

5. Let $t = \lambda x.\ (\delta\ \delta)$. Since $\vdash t : o \to \omega$, $t \in WHT_1$.

The following theorem is the main result of the paper.

Theorem 6 *Let t be a term.*

1. *$t \in SN$ iff $t \in ST$.*

2. *$t \in WN$ iff $t \in WT$ iff the left reduction of t terminates iff every quasi left reduction of t terminates.*

3. *$t \in HN$ iff $t \in HT$ iff the head reduction of t terminates iff every quasi head reduction of t terminates.*

4. *For $k \geq 1$, $t \in WHN_k$ iff $t \in WHT_k$ iff, by head reduction, t reduces to a term either in head normal form or in H_k iff, by any quasi head reduction, t reduces to a term either in head normal form or in H_k.*

3 Proof of theorem 6

3.1 The standardisation results

Some implications to be proved are immediate consequences of the standardization theorem. I recall here only the main definition and the theorem. Elementary (and very short) proofs can be found in [7], [12]. The following definition is not the usual one (for example, the one in [2]). It can be found in [12] (or, implicitly, in [7]). It is, of course, equivalent to the usual one and the proof of this equivalence is immediate.

Definition 7 *The standard reduction $\twoheadrightarrow_{st}$ is defined by the following rules :*

- *If $u \twoheadrightarrow_{st} u'$, then $\lambda x\ u \twoheadrightarrow_{st} \lambda x\ u'$.*
- *If, for all i, $u_i \twoheadrightarrow_{st} u'_i$ then $(x\ u_1\ ...u_n) \twoheadrightarrow_{st} (x\ u'_1\ ...u'_n)$.*
- *If $(a[x := b]\ \vec{c}) \twoheadrightarrow_{st} t'$ then $((\lambda x.a)\ b\ \vec{c}) \twoheadrightarrow_{st} t'$*
- *If $a \twoheadrightarrow_{st} a', b \twoheadrightarrow_{st} b'$ and for every i, $c_i \twoheadrightarrow_{st} c'_i$ then $((\lambda x.a)\ b\ \vec{c}) \twoheadrightarrow_{st} ((\lambda x.a')\ b'\ \vec{c}')$.*

Lemma 8 *Assume $t \twoheadrightarrow_{st} t'$.*

- *If t' is normal, then t reduces, by left reduction, to t'.*
- *If t' is in head normal form, then t reduces, by head reduction, to a term in head normal form.*
- *If $t' \in H_k$, then t reduces, by head reduction, to a term in H_k.*

Proof. Immediate. ■

The following result is known as the standardization theorem.

Theorem 9 *Let t be a term. If $t \twoheadrightarrow t'$, then $t \twoheadrightarrow_{st} t'$.*

Corollary 10

1. *$t \in WN$ iff the left reduction of t terminates iff every quasi left reduction of t terminates.*
2. *$t \in HN$ iff the head reduction of t terminates iff every quasi head reduction of t terminates.*
3. *For $k \geq 1$, $t \in WHN_k$ iff, by head reduction, t reduces to a term either in head normal form or in H_k iff, by any quasi head reduction, t reduces to a term either in head normal form or in H_k.*

Proof. In each case, denote by (a) (resp. $(b),(c)$) the first (resp. second, third) property. In each case, $(b) \Rightarrow (a)$ and $(c) \Rightarrow (b)$ are trivial. I only give the proofs of $(a) \Rightarrow (b)$ and $(b) \Rightarrow (c)$ in the third case. The other cases are similar.

$(a) \Rightarrow (b)$: It is enough to prove that, if t reduces to a term in H_k, then t reduces, by head reduction, to a term in H_k. The result follows immediately from theorem 9 and lemma 8.

$(b) \Rightarrow (c)$: The result is proved by induction (simultaneously for all k) on $(lg(t), cxty(t))$ where $lg(t)$ is the length of the head reduction of t to t' where either t' is in head normal form or $t' \in H_k$.

If $t = \lambda x\ u$ (because of the result is proved simultaneously for all k) or $t = (x\ \overrightarrow{u})$ the result is clear. Assume $t = ((\lambda x.a)\ b\ \overrightarrow{c})$ does not satisfy the conclusion. Then its infinite quasi head reduction is : $t \twoheadrightarrow ((\lambda x.a_1)\ b_1\ \overrightarrow{c_1}) \rightarrow (a_1[x := b_1]\ \overrightarrow{c_1}) \twoheadrightarrow ...$. Thus the reduction $(a[x := b]\ \overrightarrow{c}) \twoheadrightarrow (a_1[x := b_1]\ \overrightarrow{c_1}) \twoheadrightarrow ...$ also is quasi head and (since $lg((a[x := b]\ \overrightarrow{c})) < lg(t)$) this contradicts the induction hypothesis. ■

Another consequence is the following grammar characterization of the classes considered in definition 3.

Theorem 11 *The classes considered in definition 3 are given by the following grammars.*
$SN = (x\ \ SN\ ...\ SN) \mid \lambda x.SN \mid ((\lambda x.a)\ b\ \overrightarrow{c})$ *where* $b, (a[x := b]\ \overrightarrow{c}) \in SN$.
$WN = (x\ WN\ ...\ WN) \mid \lambda x\ WN \mid ((\lambda x.a)\ b\ \overrightarrow{c})$ *where* $(a[x := b]\ \overrightarrow{c}) \in WN$.
$HN = (x\ \Lambda\ ...\ \Lambda) \mid \lambda x\ HN \mid ((\lambda x.a)\ b\ \overrightarrow{c})$ *where* $(a[x := b]\ \overrightarrow{c}) \in HN$.
$WHN_k = (x\ \ \Lambda\ \ ...\ \ \Lambda) \mid \lambda x.WHN_{k-1}$ *(if* $k > 1$*) and* $\lambda x.\Lambda$ *(if* $k = 1$*)* $\mid$ $((\lambda x.a)\ b\ \overrightarrow{c})$ *where* $(a[x := b]\ \overrightarrow{c}) \in WHN_k$

Proof. For SN, the only non trivial thing is : If $b, (a[x := b]\ \overrightarrow{c}) \in SN$ then $t = ((\lambda x.a)\ b\ \overrightarrow{c}) \in SN$. This follows immediately from lemma 12 (1) below. This (unusual) formulation of the lemma is helpful for the next section.

The other results are immediate consequences of corollary 10. ■

Lemma 12

1. *Assume* $a, b, \overrightarrow{c} \in SN$ *and* $t = (a\ b\ \overrightarrow{c}) \notin SN$. *Then, for some* a_1, $a \twoheadrightarrow \lambda x\ a_1$ *and* $(a_1[x := b]\ \overrightarrow{c}) \notin SN$.
2. $\lambda x\ t \in SN$ *iff* $t \in SN$.
3. $(x\ t_1 ...\ t_n) \in SN$ *iff* $t_1, ..., t_n \in SN$.

Proof.

(1) Since $a, b, \overrightarrow{c} \in SN$, the infinite reduction of t looks like : $t \twoheadrightarrow ((\lambda x\ a_1)\ b_1\ \overrightarrow{c_1}\ ) \rightarrow (a_1[x := b_1]\ \overrightarrow{c_1}) \twoheadrightarrow \ldots$ The result immediately follows from the fact that $(a_1[x := b]\ \overrightarrow{c}) \twoheadrightarrow (a_1[x := b_1]\ \overrightarrow{c_1})$.

(2) and (3) are immediate. ■

3.2 Typability implies normalisation

This section is the real novelty. I prove :

Theorem 13 *1. $ST \subseteq SN$.*

2. $WT \subseteq WN$.

3. $HT \subseteq HN$

4. For $k \geq 1$, $WHT_k \subseteq WHN_k$.

The idea of the proof is the following.

To prove the strong normalization in D, I prove a substitution lemma (see lemma 18) : If t and u are typed strongly normalizing terms, then $t[x := u]$ also is strongly normalizing. This is proved by induction on a triple : first the type of u, then the length of the longest reduction of t and finally the complexity of t. The theorem follows immediately, by induction on the complexity of terms, since $(u\ v) = (x\ v)[x := u]$ where x is a fresh variable.

To prove the other results (on WN, HN, WHN_k), I define a set N_1 of triples (Γ, t, A) where Γ is a typing context, t is a term and A is a type. This set is, intuitively, a weak version of typed strongly normalizing terms. The key point is another substitution lemma which is a weak version of the one for SN and which is proved in a very similar way. The results easily follow from the fact that if $\Gamma \vdash t : A$, then $(\Gamma, t, A) \in N_1$ and this is an immediate consequence of the substitution lemma. Note that one unique substitution lemma is enough to deal with all these systems.

The following proposition should help to understand the definition of N_1 and the relation between the two substitution lemmas.

Proposition 14 *SN is characterized by the following rules. Let $t = \vec{\lambda}\ (R\ \vec{c})$ where R is either a redex or a variable.*

1. *If $R = ((\lambda x.a)\ b)$. Let R' be the reduct of R.*
   - *If x appears in a and $\vec{\lambda}\ (R'\ \vec{c}) \in SN$, then $t \in SN$.*
   - *Otherwise, if $b \in SN$ and $\vec{\lambda}\ (R'\ \vec{c}) \in SN$, then $t \in SN$*
2. *If $R = x$ and, for each $i, c_i \in SN$, then $t \in SN$.*

Proof. Immediate. ■y

Definition 15 *The set N_1 of triples (Γ, t, A) (where Γ is a typing context, t is a term and A is a type) is defined by the following rules :*

1. *If, for each j, $(\Gamma, t, A_j) \in N_1$ and $A_j \in S$, then $(\Gamma, t, \bigcap A_j) \in N_1$.*

   In the other rules, I assume $A = A_1 \to ... \to A_n \to a$ (where a is a variable or ω) i.e. $A \in S$ and $t = \lambda x_1 ... \lambda x_p\ (R\ \overrightarrow{u})$ where R is either a redex or a variable.

2. *If $a = \omega$ and $p \geq n$ then $(\Gamma, t, A) \in N_1$.*

   Otherwise :

3. *If $R = x$. Assume that, for $1 \leq i \leq k$, $(\Gamma, u_i, B_i) \in N_1$ and*

   - $\Gamma \vdash x : B_1 \to ... \to B_k \to A_{p+1} \to ... \to A_n \to a$
   - $\Gamma \vdash x_j : A_j$ *for* $1 \leq j \leq p$
     then $(\Gamma, t, A) \in N_1$.

4. *If R is a redex and $(\Gamma, t', A) \in N_1$ (where t' is the head reduct of t), then $(\Gamma, t, A) \in N_1$.*

Definition 16 *1. For $t \in SN$, $l_0(t)$ denotes the length of the longest reduction of t.*

2. For $(\Gamma, t, A) \in N_1$, $l_1(\Gamma, t, A)$ denotes the number of rules used to prove (cf. definition 15) that $(\Gamma, t, A) \in N_1$.

Examples and comments

1. Let $I = \lambda x\, x$. Then, $l = l_1(\emptyset, (I\ I), (o \to o) \cap \omega) = 4$.
   - By rule 1, $l = 1 + l_1(\emptyset, (I\ I), o \to o) + l_1(\emptyset, (I\ I), \omega)$
   - By rule 4, $l_1(\emptyset, (I\ I), o \to o) = 1 + l_1(\emptyset, I, o \to o)$
   - By rule 3, $l_1(\emptyset, I, o \to o) = 1$
   - By rule 2, $l_1(\emptyset, (I\ I), \omega) = 1$

2. It can be proved (this is sometimes called the fundamental lemma of maximality) that $l_0(t)$ is equal to the number of rules used to prove (cf. proposition 14) that $t \in SN$. This observation better shows the similarity between the two cases of lemma 18. Since I will not use this result I don't prove it.

3. It is clear that, if t reduces to t' by left reduction, then $l_1(\Gamma, t', A) \leq l_1(\Gamma, t, A)$ and the unequality is strict except if the last rule used is 2. This will be used without mention.

Lemma 17 *1. If $(\Gamma, t, A) \in N_1$ then either t is solvable or $A = A_1 \to ... \to A_n \to \omega$ and t reduces, by head reduction, to a term in H_n.*

2. $(\Gamma \cup \{x : A\}, u, B) \in N_1$ iff $(\Gamma, \lambda x\, u, A \to B) \in N_1$.

3. Assume $\Gamma \vdash x : A_1 \to ... A_k \to B$.

   (a) If, for all i, $(\Gamma, u_i, A_i) \in N_1$, then $(\Gamma, (x\ u_1 ... u_k), B) \in N_1$.

   (b) If $(\Gamma, (x\ u_1 ... u_k), B) \in N_1$ and $B \neq \omega$ then $(\Gamma, u_i, A_i) \in N_1$ for all i.

4. Let R be a redex and R' be its reduct. If $(\Gamma, (R'\ \vec{u}), A) \in N_1$, then $(\Gamma, (R\ \vec{u}), A) \in N_1$.

Proof.
- 1 and 2 are proved by induction on $l_1(\Gamma, t, A)$ and case analysis. I examine only the most significant case : case 2 (if). Let $\lambda x\ u = t \in N_1$. Assume the last rule used is 2. Thus, $\Gamma \vdash t : A_1 \to ... \to A_n \to \omega$, $t = \lambda x\, \lambda x_1 ... \lambda x_p\ (R\ \overrightarrow{v})$ and $p+1 \geq n$. It follows that $\Gamma, x : A_1 \vdash\ u = \lambda x_1 ... \lambda x_p\ (R\ \overrightarrow{v}) : B = A_2 \to ... \to A_n \to \omega$ and $p \geq n-1$. Thus $(\Gamma \cup \{x : A_1\},\, u, B) \in N_1$.
- 3 (a) is immediate. (b) : The last rule is not 4. Since $B \neq \omega$, the last rule is not 2. Then it is 1 or 3 and the result follows.
- 4 is trivial. ∎

Lemma 18 (substitution lemma) *1. Assume $t, u \in SN \cap ST$. Then $t[x := u] \in SN$.*

2. Assume $(\Gamma \cup \{x : B\},\ t, A) \in N_1$ and $(\Gamma, u, B) \in N_1$. Then $(\Gamma,\ t[x := u], A) \in N_1$.

Proof. The proofs are done by induction on $(type(u), l(t), cxty(t))$ where $l = l_0$ or l_1 according to the result we are proving. For (2), I may assume (by rule 1) that $A \in S$ and that $A \neq \omega$ (otherwise there is nothing to prove). To simplify notations, I will write $t \in N_1$ (and similarly $l_1(t)$) instead of $(\Gamma, t, A) \in N_1$ if the intended context and type is clear.

- If $t = \lambda y\ v$. The result follows from the induction hypothesis and lemma 12 (2) or 17 (2).
- If $t = (y\ v_1\ ...\ v_n)$ for $y \neq x$. The result follows from the induction hypothesis and lemma 12 (3) or 17 (3).
- If $t = ((\lambda y.b)\ c\ \overrightarrow{d})$. By theorem 11 or 17 (4) it is enough to show that $(b[x := u][y := c[x := u]]\ \overrightarrow{d}[x := u]) = t'[x := u]\ \in N$ where $t' = (b[y := c]\ \overrightarrow{d})$ and $N = SN$ or N_1 according to the result we are proving. But $l(t') < l(t)$ and the result follows from the induction hypothesis.
- If $t = (x\ b\ \overrightarrow{c})$. Let $b_1 = b[x := u]$ and $\vec{d} = \vec{c}[x := u]$.
  1. For SN. By the induction hypothesis, $b_1, \vec{d} \in SN$. By lemma 12 (1) it is enough to show that if $u \twoheadrightarrow \lambda y\ u_1$ then $t_1 = (u_1[y := b_1]\ \vec{d}) \in SN$. By the induction hypothesis and because $type(b_1) < type(u)$, $u_1[y := b_1] \in SN$ and thus, by the induction hypothesis and because $t_1 = (z\ \overrightarrow{d})\ [z := u_1[y := b_1]]$ and $type(u_1) < type(u)$, $t_1 \in SN$.
  2. For N_1. Let $\vec{c} =\ c_1\ ...\ c_p$ and $\Gamma \vdash t : A$. Then $\Gamma \vdash x : B \to C_1 \to ... \to C_p \to A$. By lemma 17 (3) $b, \vec{c} \in N_1$ and thus, by the induction hypothesis, $b_1,\, \vec{d} \in N_1$.
     - If $A = A_1 \to ... \to A_n \to \omega$ and u reduces (by head reduction) to $\vec{\lambda}\ u_1$ where the length of $\vec{\lambda}$ is at least $n+p+1$ the result is clear.

– Otherwise (by lemma 17 (1)) u is solvable. If the head normal form of u does not begin with λ the result follows immediately from lemma 17 (3). Otherwise u reduces (by head reduction) to $\lambda y\ u_1$. It is enough to show that $t_1 = (u_1[y := b_1]\ \vec{d}) = (z\ \vec{d})\ [z := u_1[y := b_1]] \in N_1$. By lemma 17 (3) $(z\ \vec{d}) \in N_1$. Since $type(u_1) < type(u)$ it is enough, by the induction hypothesis, to show that $u_1[y := b_1] \in N_1$. This follows from the induction hypothesis and the fact that : $b_1 \in N_1$ and, by lemma 17 (2) $u_1 \in N_1$ and $type(b_1) < type(u)$. ■

Corollary 19 *1. $ST \subseteq SN$.*

2. If $\Gamma \vdash t : A$, then $(\Gamma, t, A) \in N_1$.

Proof. By induction on the derivation. The only non trivial case is $t = (u\ v) = (x\ v)[x := u]$. The result follows from the induction hypothesis and lemma 18. ■

End of the proof of theorem 6

Assume $t \in WT$ (resp. HT, WHT_k). By corollary 19, $(\Gamma, t, A) \in N_1$ for some Γ, A. The result is proved by induction on $(l_1(\Gamma, t, A), cxty(t))$. Let $t = \vec{\lambda}\ (R\ \vec{u})$ where R is a redex or a variable.

- If R is a variable. For HN and WHN_k the result is clear. Otherwise we have to show that $\vec{u} \in WN$. Since $t \in WT$, $type(R) \in WD$ thus $\vec{u} \in WT$. By the induction hypothesis $\vec{u} \in WN$.

- If R is a redex. It is enough to show that $t' \in WHN_k$ (where t' is the head reduct of t). This follows immediately from the induction hypothesis (since $l_1(\Gamma, t', A) < l_1(\Gamma, t, A)$). ■

3.3 Normalisation implies typability

In this section I prove

Theorem 20 *1. $SN \subseteq ST$.*

2. $WN \subseteq WT$.

3. $HN \subseteq HT$.

4. For $k \geq 1$, $WHN_k \subseteq WHT_k$.

Proof. By induction on $(l(t), cxty(t))$ where $l(t)$ is $l_0(t)$ if $t \in SN$ and the length of the left reduction of t to its normal form if $t \in WN$ (resp. its head normal form if $t \in HN$, resp. a term in H_k if $t \in WHN_k - HN$ and $k \geq 1$).

1. If $t = \lambda x\ u$. This follows immediately from the induction hypothesis.

2. If $t = (x\ v_1\ ...\ v_n)$.

(a) For SN and WN: By the induction hypothesis, for every j, $x : A_j, \Gamma_j \vdash v_j : B_j$. Then $x : \bigcap A_j \cap (B_1, ..., B_n \to o)$, $\bigcap \Gamma_j \vdash t : o$.

(b) For HN and WHN_k $(k \geq 1)$: The v_i are given the type ω and x is given the type $\underbrace{\omega \to ... \to \omega}_{n} \to o$

3. If $t = ((\lambda x.a)\ b\ \overrightarrow{c})$. By the induction hypothesis, $(a[x := b]\ \overrightarrow{c}) \in ST$ (resp. WT, HT, WHT_k)

   (a) If x occurs in a. Let A_1 ... A_n be the types of the occurrences of b in the typing of $(a[x := b]\ \overrightarrow{c})$. Then t is typable by giving to x and b the type $A_1\ \cap ... \cap A_n$.

   (b) Otherwise

   - For SN : By the induction hypothesis b is typable of type B and then t is typable by giving to x the type B.
   - For WN, HN and WHN_k : t is typable by giving to x and b the type ω. ■

References

[1] R. Amadio, P.L. Curien. *Domains and Lambda-Calculi.* Cambridge University Press, 1998.

[2] H. P. Barendregt. *The Lambda Calculus, its syntax and semantics.* North Holland 1984.

[3] H.P. Barendregt, M. Coppo, M. Dezani-Ciancaglini. *A Filter Lambda Model and the Completeness of Type Assignment.* J. Symbolic Logic 48, 1983, 931–940.

[4] A. Bucciarelli, S. De Lorenzis, A. Piperno, I Salvo. *Some Computational Properties of Intersection Types.* LICS 1999.

[5] M. Coppo, M. Dezani. *A new type assignment for λ-terms.* Archive for Mathematical Logic 19 (1978) 139-156.

[6] M. Coppo, M. Dezani, B. Venneri. *Functional characters of solvable terms.* Z. Math. Log. Grund. Math 27 (1981) 45-58.

[7] R. David. *Une preuve simple de résultats classiques en λ calcul.* C. R. Acad. Sci. Paris, Série I, 1401-1406, 1995

[8] J. Gallier. *Typing untyped λ-terms, or reducibility strikes again.* Annals of pure and applied logic 91 (1998) 231-270.

[9] G. Huet. *Initiation au λ-calcul.* Techn. Report. Université Paris 7. 1991

[10] J. L. Krivine. *Lambda Calcul Types et Modèles.* Masson (Collection ERI) Paris 1990.

[11] D. Leivant. *Typing and Computational Properties of Lambda Expressions.* Theoritical Computer Science 44, 1986, 51–68.

[12] R. Loader. *Notes on Simply Typed Lambda Calculus.* Manuscript *1998.*

[13] R. Matthes. *Short proofs of normalization.* Manuscript 1999.

[14] S. van Bakel. *Complete Restrictions of the Intersection Type Discipline.* Theoritical Computer Science 102, 1992, 135–163.

[15] S. van Bakel. *Intersection Type Disciplines in Lambda Calculus and Applicative Term Rewriting Systems.* Ph.D. Thesis, Mathematisch Centrum, Amsterdam, 1993.